Attorney Docket No. Stubblebine 1

Inventor: Stuart Stubblebine

SPECIFYING SECURITY PROTOCOLS AND POLICY CONSTRAINTS IN DISTRIBUTED SYSTEMS

I hereby claim the benefit under 35 U.S.C. 119(e) of the U.S. provisional application filed April 30, 1996 and assigned Serial No. 06/016,788

Ins A1

FIELD OF THE INVENTION

The present invention is directed to specifying security protocols and policy constraints in distributed systems, and more particularly, to specifying security protocols and policy constraints for establishing secure channels in distributed systems using freshness constraints imposed on channel (recent-secure) authentication.

BACKGROUND OF THE INVENTION

Authentication architecture that scales globally is desirable to support authentication and authorization in electronic commerce. A characteristic of universal electronic commerce is that clients and commercial servers not previously known to one another must interact. An essential feature of an authentication service in large distributed systems is revocation. Revocation entails rescinding authentication and authorization statements that have become invalid. Revocation is needed because authentication information changes with time due to a compromise or suspected compromise of an entity's private key, a change of affiliation, or a cessation of an entity's operation. When a compromise is discovered, rapid revocation of information is required to prevent unauthorized use of resources and electronic fraud.

Revocation usually has the following properties. It should be fail-safe or assured with bounded delays, i.e., it should be definite. The mechanism for posting and retrieving updates must be highly available, and retrieved information should be recent if not current. Protection and performance trade-offs should be adjustable to suit varying risk adversity. When a compromise is discovered, delays in revocation should be decidedly bounded in time. A compromised revocation service should not allow illegitimate identification credentials to be issued.

However, there are factors which make revocation in a large distributed environment difficult. These factors include size, trust, security, the distributed nature of the system and the temporal dynamics of the system. That is, numerous entities not previously known to one another may need to interact securely. Entities have different views of the trustworthiness of intermediaries and of each other. Protection of computing environments is variable and uncertain. The authenticating entity's knowledge of authentication information can be inaccurate due to communication latency, failures, and active wiretapping. In addition, authentication and authorization information changes with time and is unpredictable.

There is little in the art focusing on revocation and validity assertions in large distributed systems. Kerberos and DCE (based on Kerberos) have been used in local autonomous realms in large distributed systems. However, shared secret crypto-systems such as these have inherent drawbacks when scaling to a large distributed system.

Authentication in large distributed systems is moving toward the integration of local network authentication servers with global directories (e.g., those based on the X.500

directory) and open authentication architectures (e.g., those based on the X.509 directory) using public key cryptography.

Global authentication architectures based on public key cryptography assume that named principals to be authenticated maintain the confidentiality of their private keys. Certificates using public key cryptography enable authentication information of the authority of the certificate contents to be distributed using servers that need not be trusted. Intermediaries, called certifiers or certification authorities (when authority is assumed by authenticating entities), create cryptographically protected statements called certificates. Identification authorities, having authority in identification of entities, issue identification certificates. Identification certificates assert that a public key is associated with an entity having a unique name. Revocation authorities, having authority on the status of certificates, issue revocation certificates. Revocation certificates assert the status of certificates previously issued. Revocation policies, which are typically included in authentication policies, represent a bounded delay before an authentication entity becomes current on the accuracy of authentication information. Authentication conforming to these policies is called recent-secure authentication. The entity or agent doing the authentication is called the authenticating entity. Propagation of certificates through servers, such as directory servers, can introduce delays.

As noted above, Kerberos is a distributed authentication service that allows a process (client) running on behalf of a principal (user) to prove its identity to a verifier (an application server or just a server) without sending data across the network that might allow an attacker or verifier to subsequently impersonate the principal. Kerberos can also provide integrity and confidentiality for data sent between the server and client. However, Kerberos does not protect all messages sent between two computers. It only protects the messages from software that have been written or modified to use it. Kerberos uses a series of encrypted messages to prove to a verifier that a client is running on behalf of a particular user. The service includes using "time stamps" to reduce the number of subsequent messages needed for basic authentication and a "ticket-granting" service to support subsequent authentication without reentry of a principal's password. It should be noted that Kerberos does not provide authorization, but passes authorization information generated by other services. Therefore it is used as a base for building separate distributed authorization services.

Recent-secure authentication is based on specified freshness constraints for statements made by trusted intermediaries (certifiers) and by principals that may be authenticated. These statements represent assertions regarding whose authenticity can be protected using a variety of mechanisms ranging from public or shared-key to physical protection. Freshness constraints restrict the useful age of statements. They can come from initial authentication assumptions and can also be derived from authentic statements which may themselves be subject to freshness constraints.

An important requirement of revocation in large distributed systems is the fail-safe property. This means that revocation is resilient to unreliable communication. Revocation mechanisms not satisfying this property can be impeded by active attacks in which the adversary prevents the reception of revocation statements. Apparent countermeasures to these attacks may not be adequate. For example, consider the technique of cascaded delegation where a delegation certificate is created as a delegation is passed to each new system. To terminate a delegation, a "tear down" order is passed down the chain. However, due to unreliable communication or a compromise of an intermediate system, the order may not fully propagate. To remedy this, it has been proposed that each intermediate delegate periodically authenticate delegates. However, periodic authentication of predecessor delegates can be vulnerable to attacks where the adversary steps down the chain blocking revocation statements until the particular link times out. The result is an additive effect on delaying revocation. Alternatively, each node could authenticate every other node at the cost of n2 messages, where n is the number of nodes. The optimal design for balancing performance and security depends on the protection of each system and the communication therebetween.

Communication latency is an inherent property of distributed systems. Consequently, authenticating entities cannot have perfect knowledge of authentication and authorization information. Therefore, failure can occur. The problem is compounded in large distributed systems. Additional certifiers represent more distributed knowledge.

Obtaining consistent knowledge of authentication data is difficult and prohibitively expensive. It is therefore necessary to quantify levels of protection that can be obtained and to reason whether they have been obtained. The practical significance of recent-secure authentication is that it enables distributed authenticating entities on a per-transaction basis to trade-off authentication costs against the level of protection.

In addition, quantifiable authentication assurances are difficult to provide if information about the intermediate system is incomplete. In spite of this, many systems operate with incomplete information. This requires the risk of operating such systems to be periodically reassessed. That is, entire industries have been dependent on reassessing shifting risks. Recent-secure authentication policies are an important variable for reassessing risk in large distributed systems. For example, proposals have been made to assign financial liability attributes to certificate authority statements in financial systems based on shifting risk.

A number of other related techniques have been proposed for effecting revocation in distributed systems. These techniques will be briefly reviewed.

With respect to certificate caches with exception notifications, authenticating entities may cache certificates and notify caches when there is a change. This approach is not well suited to large distributed systems since the notification mechanism is not fail-safe For example, an adversary could selectively block an exception notification. Also, it does not scale well if the destination caches need to be tracked. However, emergency notifications can augment a fail-safe scheme to possible shorten revocation delays provided messages reach their destinations sooner than the time-out periods. A distributed multicast infrastructure could alleviate the load on servers for distribution of notifications.

With respect to certificates having expiration times, a common technique for bounding the delay of revocation is placing explicit expiration times within certification. Statements using expiration times satisfy the fail-safe property provided that a certifier has not been compromised. Since authentication can depend on trusted intermediaries, an entity might be vulnerable to illegitimate statements made by a compromised certifier. Consequently, neither the certifier nor the authenticating entity can be assured that it or its subordinates have not been cloned due to a compromise of an arbitrary certifier that is trusted by an authenticating entity.

On-line servers and quorum schemes have been proposed whereby entities issue queries in an authenticated exchange to learn the validity of authentication/authorization information. Use of on-line servers may be justified in architectures where the server is local to the source or destination. However, network failures can significantly impact the availability of such servers for geographically distributed clients.

Replicating trusted servers for increasing availability inherently increases the risk of compromising the secret keys held by the server. Secret sharing techniques can improve availability and security, but they do so at the expense of considerable communication costs and increased delay. For example, the effective time of the statement from the quorum might be the earliest statement time in a final round used to make the decision. Due to communication costs and increased delay, geographic distribution of secret sharing servers, for the purposes of surviving network failures, may not be practical for most applications.

With respect to long-lived certificates and periodic revocation statements, revocation methods have been proposed where authorities issue long-term identification certificates and periodically publish time stamped revocation certificates. Revocation certificates can be distributed to the authenticating entity through untrusted communications.

The scalableness of this approach depends on whether servers are replicated. However, replicating the trusted identification authority inherently decreases security. In this case, a compromised server may enable an adversary to issue new identification certificates.

With respect to off-line identification authority and on-line revocation authority, an approach for increasing the availability and security of an authentication service calls for joint authorities. An off-line identification authority generates long-term certificates and an on-line revocation authority creates countersigned certificates with short lifetimes. The effective lifetime is the minimum lifetime of both certificates.

The joint authority approach benefits from the fact that the compromise of the on-line server does not enable the adversary to issue new identification certificates. As expected, a compromised revocation authority could delay revocation until the authority of the on-line server expires. However, the period of compromise may be extended further if the revocation authority issues revocation certificates with longer lifetimes.

An alternative approach to creating countersigned certificates is to authenticate a channel to an on-line (trusted) database server and to retrieve original certificates. However, authenticated retrieval of certificates alone may be insufficient to provide adequate assurance that a certificate is current. For example, when providing high availability for geographically distributed clients, the revocation service might replicate the database and use optimistic consistency control techniques. These techniques do not guarantee the consistency of stored certificates at the time of retrieval. Consequently, the presence of a certificate in a local replica might represent stale information. Additional delays occur as certificates are exported to trusted subsystems for local retrieval. Also, the on-line revocation server/database is subject to the scaling limitations inherent to on-line servers as discussed above.

As set forth above, neither Kerberos nor any other authentication system focuses on revocation and validity of assertions in large distributed systems. These prior systems all have inherent problems regarding revocation in complex systems.

SUMMARY OF THE INVENTION

The present invention provides improved system security in distributed systems by making precise what degree of protection is desired and obtained. Freshness constraints and time stamped certificates are employed to enforce revocation. In addition, recent-secure policies associated with statements of authentication liability are provided based on axioms and principals.

The present invention provides system security in distributed systems by making authentic statements by trusted intermediaries, deriving freshness constraints from initial policy assumptions and the authentic statements, and imposing freshness constraints to effect revocation. Revocation can be arbitrarily bounded by adjusting the freshness constraints and applying certain principles.

Generally, revocation is enforced by distinguished principals issuing initial assertions and by asserting the validity of the initial assertions at a time specified by a time stamp. The distinguished principals assert one or more principals with authority for asserting a time stamped validity assertion. Freshness constraints are asserted indicating a length of time. A relation is verified for each assertion necessary to verify a secure channel.

In addition, for a more specific example, system security in distributed systems is provided by using identification authorities for issuing long-lived identification certificates, using revocation authorities for issuing time stamped certificates, and posting updates to the revocation authorities. The identification authorities also specify freshness constraints for revocation within an identification certificate. The revocation authority can be the identification authority.

Another method for enforcing revocation includes designating a policy certification authority for dictating policy to subordinates, designating an organization for receiving policy from the policy certification authority, designating entities within the organization, issuing a delegation certificate specifying an entity authorized as an authority for issuing time stamped revocation certificates, and issuing time stamped revocation certificates from the authority. The time stamped revocation certificates are issued in an unidirectional manner. The authority can use freshness constraints when issuing the time stamped revocation certificates.

A revocation system can include an identification authority for issuing long-lived identification certificates and a revocation authority for issuing time stamped revocation certificates, the identification authority and the revocation authority having only unidirectional communication to a network. The identification authority specifies freshness constraints for revocation within the identification authority and the revocation certificates meet the specified freshness constraints.

Further, a revocation system for use in a distributed system includes a policy certification authority for dictating policy, an organization dictated to by the policy certification authority and issuing identification certificates, entities for receiving the identification certificates from the organization, delegation certificates containing information specifying an entity as an authority for issuing time stamped revocation certificates, and a revocation authority for issuing the time stamped revocation certificates to the entity.

The revocation authority can provide the time stamped revocation certificates at a predetermined time or on demand. The system can also include a storage location for obtaining certificates. The storage location can include a replicated directory having varying levels of persistent storage. The replicated directory includes a high level directory for frequently replicating information, a medium level directory for often replicating information, and a low level directory for infrequently replicating information. The high level directory includes the time stamped revocation certificates, the medium directory includes the time stamped revocation certificates and the delegation certificates, and the low level directory includes the time stamped revocation certificates, the delegation certificates, and the identification certificates.

These and other objects, advantages, and salient features of the present invention will become apparent from the following detailed description which, when taken in connection with the annexed drawings, discloses preferred but non-limiting embodiments. In the drawings, like reference numerals refer to like parts throughout.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an architectural diagram of a prior art computer network within which the invention can be applied.

Fig. 1A is an architectural diagram of another prior art computer network within which the invention can be applied.

Fig. 2A is an architectural diagram showing an example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 0930".

Fig. 2B is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 0945".

Fig. 2C is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1000".

Fig. 2D is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1015".

Fig. 2E is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1016".

Fig. 2F is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1017".

Fig. 2G is an architectural diagram showing an example of the implementation of a modified embodiment of the invention in the prior art computer network of Fig. 1, at the same stage as in Fig. 2D when the network clock is "3/12/97 @ 1015".

Fig. 2H is an architectural diagram showing the example of the implementation of the modified embodiment of the invention in the prior art computer network of Fig. 1, at the stage following Fig. 2G, when the network clock is "3/12/97 @ 1016".

Fig. 3A is an architectural diagram showing an example of the implementation of a second embodiment of the invention in the prior art computer network of Fig. 1A, showing identification authority server computer 102 issuing identification certificate 220 when the network clock is "2/2/96 @ 0000".

Fig. 3B is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing identification authority server computer 102 issuing the identification certificate 223 when the network clock is "2/15/97 @ 0400".

Fig. 3C is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing revocation authority pointer computer 108 issuing medium-term delegation certificate 230 when the network clock is “3/8/97 @ 1100".

Fig. 3D is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing revocation authority server computer 106 issuing the time stamped validity certificate 225 when the network clock is “3/12/97 @ 0940".

Fig. 3E is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing user computer 130 accessing certificates 223, 225, and 230 from network storage 116 in preparation for making an access to the file server 140 when the network clock is “3/12/97 @ 0945".

Fig. 3F is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing the user computer making an access request to the file server 140 when the network clock is “3/12/97 @ 0946".

Fig. 3G is an architectural diagram showing an example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing the verification authority server of the file server computer 140 granting the access request of the user computer 130 when the network clock is “3/12/97 @ 0947".

Fig. 4A is a flow chart explaining the present invention using axioms.

Fig. 4B is a flow chart explaining the present invention using policy constraints and assumptions of the revocation service according to the present invention.

Fig. 5A is a diagram of a basic certificate topology according to the present invention.

Fig. 5B is a diagram of a certificate topology according to a first embodiment of the present invention.

Fig. 5C is a diagram of a certificate topology according to a second embodiment of the present invention. and

Fig. 6 is a diagram of a replicated directory with storage which can be used with the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The concept of specifying security protocols and policy constraints for establishing secure channels in distributed systems will be explained in conjunction with related systems. In particular, recent-secure authentication can be related to a hybrid optimistic/pessimistic method of concurrency control that allows for the selective application of the most efficient method characterized by the probability of conflict. For example, small freshness intervals correspond to a pessimistic method requiring more expensive mechanisms than those required by larger freshness intervals.

To effect revocation, authenticating entities impose freshness constraints, derived from initial policy assumptions and authentic statements made by trusted intermediaries, on credentials or authenticated statements made by the trusted intermediaries. If freshness constraints are not presented, then the authentication is questionable. By adjusting freshness constraints the delay for certain revocation can be arbitrarily bounded. This is important since zero delays can not be achieved in large distributed systems due to communication latency. Freshness constraints within a certificate enables the design of a secure and highly available revocation service.

Before describing the present invention in detail, the theory for specifying temporal features of statements made by entities and rules for reasoning about them are presented. The primary feature of the present invention is attaining recent-secure authentication rather than specifying implementation algorithms for maintaining recent-secure channels.

Performance optimizations and increased assurance can be realized if policies for querying entities can be refined and the reasoning behind the policies is formally analyzed. Recent-secure authentication is necessary both during initial authentication as well as during a session since revocation can occur at any time.

Principals can be used as the basis for specifying freshness constraints and can be classified according to freshness classes. Freshness constraints can be in identification certificates, separate certificates, or specified by whoever is an authority. An authority is an entity who assumes risk. A freshness constraint associated with a principal can depend on an identification authority's certificate issuance policy since this policy specifies security relevant information such as the randomness of keys, software and hardware protection for key management, identification, and authentication requirements for certificate registration. It should be noted that many policies may exist within an identification authority's certificate. A brief review of the theory of secure channels will now be given.

System specification includes specifying system entities stating assumptions about synchronization bonds between clocks of principals, annotating initial assumptions and messages using mathematical relationship statements and interpreting certificate policies as constraints on axiomatic derivations. All entities in a system are referred to as principals. A distinguished principal is an authenticating entity which authenticates a channel. Basic named principals are entities that cannot say things directly on a computer. For example, they can be people i.e., *Bob,* groups of people or roles such as *"identification authority."* Channels or channel principals are principals that can say things directly. An I/O port is an example of a channel that reports the receipt of a message. A key i.e., K_{Bob} and cryptographic algorithm is an example of a channel that makes cryptographic statements. Cryptographic channels are of primary interest when communication transits untrusted domains and is subject to wiretapping.

Initial assumptions are messages annotated as formulas using "speaksfor" and "says" statement. If K_{Bob} and *Bob* are principals then $K_{Bob} \Rightarrow Bob$ *is* a statement. The symbol $\Rightarrow$ "speaksfor" relation. Suppose K_{Bob} is a channel and *Bob* is a named principal, then the above statement allows one to deduce that the channel K_{Bob} represents *Bob.*

If *IA* is a principal and *s* is a statement then *IA* says s is also a statement. If *"IA says s"* one can proceed as though *IA* is willing to say s. It does not necessarily mean that IA had actually articulated the statement. For example, one may have derived this from the axioms. It should be noted that a principal could be lying.

For acceptable performance, basic channel principals can have clocks that are loosely synchronized to an external time reference or synchronized with other clocks. Clocks can be used for annotating message time stamps. Synchronization bounds on clocks of principals are specified as initial assumption of channel principals. Assumptions about synchronization bounds can be represented as:

$$/\textit{clock}\ (P_1,t) = \textit{clock}\ (P_2,t)/\leq \textit{Synchronization bound}\ (P_{1,}P_2)$$

where clock (P,t) represents the time on P's clock at the real time *t* and Synchronization bound *(P_1,P_2)* represents the synchronization bound. The channel's synchronization accuracy is known to other principals and the authenticating entity. Loosely synchronized clocks are used to expire keys and to age statements. The accuracy of clock synchronization constrains the granularity for aging statements and expiring keys. Granularity constraints on expiring keys are not a practical problem in situations where sufficiently pessimistic assumptions can be made by assigning key lifetimes. However, granularity is an issue for fail-safe revocation since the practical bound on revocation may need to be on the order of tens of minutes or less. The reliance on clock synchronization for refreshing statements makes recent-secure authentication susceptible to vulnerabilities due to clock failures. Therefore, the assumption of synchronized clocks must be carefully scrutinized.

The present invention assumes that statements from each channel principal can be ordered and missing statements can be detected. In practice, this requirement can be carefully relaxed. For example, if statements can be ordered and each statement provides a complete interpretation, then interpretation of missing statements may be unnecessary. Also, the order of statements from different sources can be established using clock synchronization and external synchronization accuracies can be determined.

A principal may also assert a statement and a time attribute. For example,

K_{IA} says s at t.

It is not necessarily a fact in this statement that a principal says s at time t. As mentioned above, a principal could be lying. However, this axiom captures the notion that if one trusts a principal and is able to discern a statement made by it, then one can also trust the facts concerning the statement's time attribute. Not all "says" statement's have time attributes.

"Speaksfor" relations are given time constraints. The time constraints are specified using a «notbefore» and "notafter" suffix. For example, the statement

$K_{Bob} \Rightarrow Bob$ notbefore t_1 notafter t_2

indicates that during the closed interval $[t_1, t_2]$, $K_{Bob} \Rightarrow Bob$.

For analysis purposes, individual messages such as public key certificates are first interpreted as statements. Then, it must be determined whether there has been an unambiguous interpretation. If so, the axioms related to the present invention are employed. Examples will now be given with respect to interpreting authentication message types (certificates) and formalizing them as statements.

In practice, an identification certificate (sometimes referred to as a certificate) contains identifying information of an entity together with its associated keying material and possibly other information such as an expiration date. The identification certificate is cryptographically protected by using the key of an identification authority. The interpretation of an X.509 and Internet Society's (X.509 compliant) IPRA identification certificate is represented as

$$\boldsymbol{K_{IA}} \textbf{ says } \boldsymbol{(K_{IA/Bob} \Rightarrow IA/Bob} \textbf{ notbefore } \boldsymbol{t_1}\textbf{, notafter } \boldsymbol{t2)}$$

K_{IA}, which is the identification authority's key, asserts the ⇒ relationship between ($K_{IA/Bob}$ and *A/Bob* between the validity interval $[t_1, t_2]$. Since a time stamp is not in the identification certificate, the time is annotated. Consequently, from the above statement, the authentication entity cannot determine the recentness of the statement.

Where the identification authority can be adequately protected, architectures may use identification certificates as the primary basis for determining freshness. Since none of the current standards specify a time stamp within the identification certificate, the X.509 certificate format with a time stamp is assumed as follows:

$$\mathbf{K_{CA}} \textbf{ says } \mathbf{(K_{CA}} \Rightarrow \boldsymbol{Bob} \textbf{ notbefore } \boldsymbol{t_1} \textbf{ notafter } \boldsymbol{t_2}\mathbf{)} \textbf{ at } \boldsymbol{t_3}\mathbf{.}$$

A certificate revocation list (CRL) or revocation certificate indicates what certificates have changed in "status." The interpretation of a revocation certificate typically has the format of the time stamped identification certificate. The interpretations of the revocation certificates can be summarized as follows.

The absence of a referenced identification certificate implies that it is "current" at the time of the revocation certificate. This interpretation asserts the interpretation of the referenced identification certificate with the time of the revocation certificate.

When the referenced certificate is invalid or is about to be invalid it is referred to as "revoked." This interpretation reasserts the original statement specifying a "notafter" time corresponding to the revocation date. If the time stamp is present the time of the statement is annotated.

When the certificate binding is extended to a new time it is referred to as being «extended." This interpretation reasserts the original statement and the "notafter" time is set for the new time. If the time stamp is present the time of the statement is annotated. If the certificate is temporarily suspended, it is referred to as being "suspended." This interpretation consists of two statements: a statement indicating a «notafter" time set sooner than the original certificate; and a new statement with a "notbefore" time set at a later time. If a time stamp is present, the time of the statement is annotated.

Initial assumptions and messages as used in the present invention as formulas will now be annotated using "speaks for" and "says" statements. Some are repeated for clarity and completeness.

The following syntax will be used in annotating formulas. The notation $\vdash s$ means that s is an axiom of the theory or is provable from the axioms. That is, the truth of s is an assumption and can be derived. The symbol $\supset$ means implies. The symbol $\Rightarrow$ is the «speaksfor" relation between principals. The symbol t_{now} represents the time of verification.

Formulas are defined inductively, as follows. A countable supply of primitive propositions *pO, pl, p2*, . . . are formulas. If s and s' are formulas then so are $\neg s$ and $s \wedge s'$ If A and B are principal expressions the following are formulas:

$A \Rightarrow B$ **notbefore** t_1 **notafter** t_2**; and**

A **says** s **at** t**.**

Statements are inductively defined according to the following axioms. The axioms for statements are as follows:

(S1) If s is an instance of a propositional-logic tautology then $\vdash s$.

(S2) If $\vdash s$ and $(\vdash s \supset s')$ then $\vdash s'$.

(S3) $\vdash$ (A **says** $(s \supset s')$ **at** $t) \supset ((A$ **says** s **at** $t) \supset A$ **says** s **at** t)).

(S4) If $\vdash s$ then $\vdash A$ **says** s **at** t_{now}, for every principal A. From (S1)-(S4) the following theorem is obtained.

(S5) $\vdash A$ **says** $(s \wedge s')$ **at** $t \equiv (A$ **says** s **at** $t \wedge A$ **says** s **at** t).

The axioms for principals are as follows:

(P1) $\vdash \wedge$ is associative, commutative and idempotent.

(P2) $\vdash |$ is associative.

(P3) $\vdash |$ distributes over $\wedge$ in both arguments.

Additional axioms for principals include the time of a statement made by $(A \wedge B)$ is equivalent to both A and B saying it at the same time.

(P4) $\vdash (A \wedge B)$ **says** s **at** $t \equiv (A$ **says** s at $t) \wedge (B$ **says** s **at** t).

B quoting A at time t has the definition:

(P5) $\vdash$ (B|A) says s at $t \equiv B$ **says** $(A$ **says** s **at** $t)$ **at** t.

The following new axiom is introduced in (P6). This axiom allows more restrictive relations to be obtained from less restrictive ⇒ relations. This axiom is used to normalize suffix constraints prior to applying other rules such as the transitivity of ⇒.

(P6) $\vdash$ ($A \Rightarrow B$ **notbefore** t_1 **notafter** t_2) $\supset$ ((($t_1 \leq t_3$) $\wedge$ ($t_4 \leq t_2$)) $\supset$ $A \Rightarrow B$ **notbefore** t_3 **notafter** t_4).

In addition:

(P7) $\vdash$ ($A \Rightarrow B$ **notbefore** t_1 **notafter** t_2) $\supset$ (((**A says s at** t_3) $\wedge$ ($t_1 \leq t_{now}.t_3 \leq t_2$)) $\supset$ (B says s at t_3)).

The ⇒ relation in (P7) allows removal of a level of indirection. The constraint $t_1 \leq t_3 \leq t_2$ is meaningful if the principals are trusted not to lie. For example, revocation authorities are trusted not to lie when specifying the time of revocation certificates. This hand-off axiom (P7) allows principals to derive new facts as follows:

(P8) $\vdash$ (**A says (**$B \Rightarrow A$ **notbefore** t_1 **notafter** t_2**) at** t_3) $\supset$ ($B \Rightarrow A$ **notbefore** t_1 **notafter** t_2).

Fig. 1 is an architectural diagram of a prior art computer network within which the invention can be applied. The prior art network of Fig. 1 includes a public network 100 and a private network 103. The public network 100 includes a computer 130, a file server computer 140, a network clock 150, and network storage devices 112 and 116. The private network 103 includes computer 102 that has an input 107, computer 104 that has an input 105, and computer 106. Computers 102 and 106 are also connected to the public network 100.

Fig. 2A is an architectural diagram showing an example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock 150 is “3/12/97 @ 0930". Fig. 2A includes the public network 100 and the private network 103. The public network 100 includes the computer 130 operated by a user. The file server computer 140 is programmed, in accordance with the invention, as a verification authority server. The private network 103 includes the computer 102 programmed, in accordance with the invention, as an identification authority server. User access requests are applied at input 107. The computer 104 is programmed, in accordance with the invention, as a security policy server and the input 105 provides network security policy inputs. The computer 106 is programmed, in accordance with the invention, as a revocation authority server. Computers 102 and 106 are connected for one way information flow toward the public network 100.

When computer 104 receives a network security policy input at the input 105 in Fig. 2A, the policy input can be directed for example, to establishing a long term identity of the user in the network and establishing the user’s short term security status. The security policy server of computer 104 prepares a long term policy message 200 that it sends to computer 102 and it prepares a short term policy message 202 that it sends to computer 106. The identification authority server of computer 102 prepares an identification certificate 210 to identify the user, in response to the long term security policy 200 and the user request input at 107. The identification certificate 210 includes, for example, the user name “JOHN", the security level “TOP SECRET", the expiration date of the security level “12/31/97", the freshness constraint period “20 MINUTES", the user’s public key “ABC123", and the address of the revocation authority “SERVER 106". The identification certificate 210 also includes a digital signature 211 that uniquely identifies the identification authority server computer 102 as the source of the certificate 210 and also provides for verifying the integrity of the contents of the certificate 210. (For brevity, we assume that computer 140 has the key to verify the signature on certificate 210.) The identification certificate 210 is sent to the user's computer 130. The identification certificate 210 can also be sent to the network storage 116.

The security policy server of computer 104 of Fig. 2A prepares a short term policy message 202 that it sends to computer 106. The revocation authority server of computer 106 prepares a time stamped validity certificate 215, which is updated at specified intervals. The validity certificate 215 certifies at the stated instant, the security status of a particular user and of a particular identification certificate 210 of that user. Changes in the security status stated in the validity certificates 215 are made by the revocation authority server computer 106 in response to updates in the short term security policy 202. The validity certificate 215 includes the time stamp "3/12/97 @ 0930" for example, the user's identity and the identity of the corresponding identification certificate "JOHN (211)", and the current security status "OK". The validity certificate 215 also includes a digital signature that uniquely identifies the revocation authority server computer 106 as the source of the validity certificate 215 and also provides a for verifying the integrity of the contents of the validity certificate 215. The validity certificate 215 is sent to the network storage 116, to be available to nodes in the public network 100 that need to know the current security status of the user.

Fig. 2B is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 0945". When the user at computer 130 wants to access a file from the file server computer 140, the user requests a copy of the most recent validity certificate 215 from the network storage 116. The time stamped validity certificate 215 returned to the user bears a time stamp of "3/12/97 @ 0930".

Fig. 2C is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1000". In this example, the revocation authority computer 106 provides an updated validity certificate 215' about the user at the time "3/12/97 @ 1000" in response to the updated short term policy 202' input from the security policy server computer 104. The user's new status has changed to "NOT OK". The new validity certificate 215' includes the time stamp "3/12/97 @ 1000" for example, the user's identity and the identity of the corresponding identification certificate "JOHN (211)", and the current security status "NOT OK". The updated validity certificate 215' is stored in the network storage 116.

Fig. 2D is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1015". At this time, the user's computer 130 begins its access request to the file server computer 140 programmed as a verification authority server. A copy of the user's identification certificate 210 and a copy of the first time stamped "0930" validity certificate 215 are sent to the file server computer 140 programmed as a verification authority server. The verification authority server programmed computer 140 examines the identification certificate 210 and determines that the freshness constraint period is "20

MINUTES". This means that the time stamp of the validity certificate 215 sent by the user must not be older than 20 minutes before the current instant of "1015". Since the time stamp of the validity certificate 215 sent by the user is "0930", the first validity certificate 215 is too old. The verification authority server programmed computer 140 has several options in response to this. For example, it can request a copy of the latest validity certificate 215' from the network storage 116, as is shown in Fig. 2E. Or, alternately, it can require the user's computer 130 to request a copy of the latest validity certificate 215' from the network storage 116, as is shown in Fig. 3B.

Fig. 2E is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1016". The verification authority server programmed computer 140 requests a copy of the latest validity certificate 215' from the network storage 116, which has a time stamp of "3/12/97 @ 1000". The user's new status has changed to "NOT OK". The new validity certificate 215' includes the time stamp "3/12/97 @ 1000", the user's identity and the identity of the corresponding identification certificate "JOHN (211)", and the current security status "NOT OK".

Fig. 2F is an architectural diagram showing the example of the implementation of an embodiment of the invention in the prior art computer network of Fig. 1, when the network clock is "3/12/97 @ 1017". The verification authority server programmed computer 140 denies the user's access request because the new status is "NOT OK" in the certificate 215'.

Fig. 2G is an architectural diagram showing an example of the implementation of a modified embodiment of the invention in the prior art computer network of Fig. 1, at the same stage as in Fig. 2D when the network clock is “3/12/97 @ 1015".

Fig. 2H is an architectural diagram showing the example of the implementation of the modified embodiment of the invention in the prior art computer network of Fig. 1, at the stage following Fig. 2G, when the network clock is “3/12/97 @ 1016". In this example the verification authority server programmed computer 140 denies the user’s access request because certificate 215 is too old for the freshness constraint period. The verification authority server programmed computer 140 requires the user’s computer 130 to request a copy of the latest (or an adequately recent) validity certificate 215' from the network storage 116, and to then forward it to the computer 140.

Fig. 1A is an architectural diagram of another prior art computer network within which the invention can be applied. The prior art computer network of Fig. 1A differs from the prior art computer network of Fig. 1 in that it adds an additional computer 108 that is connected to the private network 103 and to the public network 100. Computer 102 is not connected to network 103 and computer 106 is not connected to network 103. Computer 104 has a direct connection to computer 102 through input 107 and computer 104 has a direct connection to computer 106 through input 101.

Fig. 3A is an architectural diagram showing an example of the implementation of a second embodiment of the invention in the prior art computer network of Fig. 1A, showing identification authority server computer 102 issuing identification certificate 220 when the network clock is “2/2/96 @ 0000". The identification certificate 220 is issued in response to the long term security policy 200 output from the security policy server computer 104 over direct connection 107 to computer 102, which is in response to the network security policy 105. The second embodiment of the invention adds the feature of delegation of authority by using a delegation certificate. This feature is useful, for example, when it is anticipated that changes may need to be made in the types of authorizations specified in the long term identification certificate 210 of Fig. 2A. For example, the long term identification certificate 210 of Fig. 2A specifies the identity of the revocation authority as “server 106". However, the network security policy 105 may want to change the designation of the revocation authority and/or freshness constraint period before the end of the long term. In the first embodiment of Fig. 2A, this would require the identification authority server computer 102 to issue a revised identification certificate 210 or to defer the specification of the revocation authority and/or freshness constraint period and the specification of these in a second, shorter term certificate. Instead, the second embodiment of the invention of Fig. 3A provides for delegating the designation of the revocation authority to the revocation authority pointer computer 108. The identification certificate 220 of Fig. 3A sent from the identification authority server computer 102 to the user computer 130, specifies the revocation authority as = “Delegated to 108". In another example, the long term identification certificate 210 of Fig. 2A specifies the freshness constraint as a period = "20 Minutes”. However, the network security policy 105 may want to change the designation of the freshness constraint before the end of the long term. Instead of requiring the identification authority server computer 102 to issue a revised identification certificate 210, the second embodiment of the invention of Fig. 3A provides for delegating the designation of the freshness constraint to the revocation authority pointer computer 108. Identification certificate 220 of Fig. 3A specifies the freshness constraint period as = “Delegated to 108". The identification certificate 220 also includes a digital signature 224 that uniquely identifies the identification authority server computer 102 as the source of the certificate 220 and also provides for verifying the integrity of the contents of the certificate 220. (For brevity, we assume that computer 140 has the key to verify the signature on certificate 220.) The identification certificate 220 is sent to the user’s computer 130. The identification certificate 220 can also be sent to the network storage 116.

Fig. 3B is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing identification authority server computer 102 issuing the identification certificate 223 when the network clock is “2/15/97 @ 0400". The second identification certificate 223 provides the public key = “XYZ456" of the revocation authority pointer server computer 108. This public key is used to verify the digital signature 231 of the medium-term delegation certificate 230 issued by computer 108 in Fig. 3C. This second identification certificate 223 of Fig. 3B also includes a medium term duration specification of its validity, from not before “3/1/97" until it expires on “4/30/97". The certificate 223 can be stored on the network storage 116 so as to be accessible by the nodes in public network 100. The certificate 223 can be automatically issued on a periodic basis, such as once every two months, to enable verifying the digital signature in the medium-term delegation certificate 230 issued by computer 108 in Fig. 3C. If the public key does not change from one period to the next, the new certificate indicates that the public key is still valid for the next time period.

Fig. 3C is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing revocation authority pointer computer 108 issuing medium-term delegation certificate 230 when the network clock is “3/8/97 @ 1100". The medium-term delegation certificate 230 is issued in response to the medium term security policy 201 output from the security policy server computer 104, which is in response to the network security policy 105. The certificate 230 can be automatically issued on a periodic basis, such as once every week, until the policy 105 is changed. The medium-term delegation certificate 230 of Fig. 3C references “all users with level = top secret.” The second embodiment of the invention provides for delegating the designation of the revocation authority (but not freshness constraint period) to the revocation authority computer 106. The medium-term delegation certificate 230 of Fig. 3C specifies the revocation authority as = “Server 106". In accordance with the invention, this designation can be changed using revocation authority pointer server computer 108, without requiring the identification authority server computer 102 to issue a revised identification certificate 220. Also, the second embodiment of the invention provides for delegating the designation of the freshness constraint to the revocation authority pointer computer 108. The medium-term delegation certificate 230 of Fig. 3C specifies the freshness constraint period as = “20 Minutes". In accordance with the invention, this designation can be changed without requiring the identification authority server 102 to issue a revised identification certificate 220. The medium-term delegation certificate 230 of Fig. 3C also includes the medium term duration of its validity, from not before “3/10/97" until it expires on “3/17/97". The medium-term delegation certificate 230 also includes a digital signature 231 that uniquely identifies the revocation authority pointer server computer 108 as the source of the medium-term delegation certificate 230 and also provides for verifying the integrity of the contents of the certificate 230. The medium-term delegation certificate 230 is sent to the network storage 116.

Fig. 3D is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing revocation authority server computer 106 issuing the time stamped validity certificate 225 when the network clock is “3/12/97 @ 0940". The time stamped validity certificate 225 is issued in response to the short term security policy 202 output from the security policy server computer 104 over line 101 to computer 106, which is in response to the network security policy 105. The short term security policy 202 may authorize the continued issuance of time stamped validity certificates 225 with a status = "OK" until policy 202 is preempted by a subsequent short term security policy that changes that status. The revocation authority server of computer 106 prepares the time stamped validity certificate 225, which is updated at specified intervals or on demand from requesting nodes in the network 100. The validity certificate 225 certifies at the stated instant, the security status of a particular user and of a particular identification certificate 220 of that user. The validity certificate 225 can reference multiple identification certificates 220 of that user. Changes in the security status stated in the validity certificates 225 are made by the revocation authority server computer 106 in response to updates in the short term security policy 202. The validity certificate 225 includes the time stamp "3/12/97 @ 0930" for example, the user's identity and the identity of the corresponding identification certificate "JOHN (224)", and the current security status "OK". The validity certificate 225 also includes a digital signature 226 that uniquely identifies the revocation authority server computer 106 as the source of the validity certificate 225 and also provides for verifying the integrity of the contents of the validity certificate 225. The validity certificate 225 is sent to the network storage 116, to be available to nodes in the public network 100 that need to know the current security status of the user.

Fig. 3E is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing user computer 130 accessing certificates 223, 225, and 230 from network storage 116 in preparation for making an access request to the file server 140, when the network clock is "3/12/97 @ 0945".

Fig. 3F is an architectural diagram showing the example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing the user computer making an access request to the file server 140 when the network clock is "3/12/97 @ 0946". The user computer 130 sends identification certificates 220 and 223, medium-term delegation certificate 230, and time stamped validity certificate 225 to the file server computer 140. The file server computer 140 is programmed, in accordance with the invention, as a verification authority server. In the verification of these certificates, computer

140 verifies that all "not before" and expiration time constraints are satisfied at the instant of verification of "3/12/97 @ 0946" in Fig. 3F. The verification authority server programmed computer 140 examines the identification certificate 220 and uses the public key of computer 102 to verify and determine that the designation of the freshness constraint period and revocation authority has been delegated to the revocation authority pointer server 108. The verification authority server programmed computer 140 verifies the signature of identification certificate 223 using the public key of computer 102 and extracts the public key "XYZ456" from the identification certificate 223. The verification authority server programmed computer 140 uses the public key "XYZ456" from the identification certificate 223 to verify the digital signature 231 of the medium-term delegation certificate 230 issued by server 108. The verification authority server programmed computer 140 examines the medium-term delegation certificate 230 issued by server 108 and determines that the designation of the freshness constraint period is “20 MINUTES”. This means that the time stamp of the validity certificate 225 sent by the user must not be older than 20 minutes before the current instant of 3/12/97 @ 0946. The verification authority server programmed computer 140 examines the medium-term delegation certificate 230 issued by server 108 and determines that the designation of the revocation authority is = "server 106". The verification authority server programmed computer 140 examines the time stamped validity certificate 225 issued by revocation authority server computer 106 and determines that the time stamp = "3/12/97 @ 0930" satisfies the freshness constraint that it is not older than 20 minutes before the current instant of 3/12/97 @ 0946.

Fig. 3G is an architectural diagram showing an example of the implementation of the second embodiment of the invention in the prior art computer network of Fig. 1A, showing the verification authority server of the file server computer 140 granting the access request of the user computer 130 when the network clock is "3/12/97 @ 0947".

Fig. 4A is a flow chart explaining the present invention using axioms. First an authentic statement is made. The statement is made, for example, by a trusted intermediary. It is then determined whether freshness constraints are included in the authentic statement. If so, suffix constraints of the freshness constraints are normalized using statement (P6). Then, access is maintained or revoked in accordance with the freshness constraints using statements (P7) and (P8).

Now, recent-secure authentication and bounds on revocation specified as freshness constraints on statements made by entrusted entities will be explained. The recent-secure authentication will be discussed with respect to Kerberos as an example, but is not limited to a Kerberos system.

To effect revocation, authenticating entities are assumed to contain freshness constraints as set forth above, which they follow. These constraints can come from the authentication entity's participation in a distributed system. For example, a vendor conducting electronic commerce on a public network authenticates customers according to the policies of organizations taking financial liability for the transaction. Authenticating entities are given a set of credentials or statements for channel authentication. They also begin with a set of freshness constraints that embody, in part, the revocation policies. Revocation policies are also embodied in freshness constraints recommended by intermediary certifiers. That is,

Definition A⇒B *satisfies freshness constraint* $\delta_{A\Rightarrow B}$ *at time* t_{now} *iff* $A \Rightarrow B$ **notbefore t** *where* $(t_{now} - \delta_{A\Rightarrow B}) \leq t$ and $\delta_{A\Rightarrow B} \leq t_{now}$.

This defines a time constraint such as one hour, etc. Verification policies can be interpreted as constraints on axiomatic derivations in a number of ways. For example, freshness constraints can be applied to ⇒ relations during interpretation by replacing existing less restrictive "notbefore" qualifiers with more restrictive freshness constraints in appropriate "speaksfor" (sub)statements. This implies that the interpretation of messages of the specification may change due to the interpretation of verification policies. For example, if the freshness constraint $\delta_{CA\Rightarrow CA/Bob}$ = 30 minutes is assumed by the authenticating entity, and *CA* . *CA/Bob* is given, then *CA* . *OA/Bob* might be replaced with the following which is more restrictive:

CA . *CA/Bob* **notbefore** (t_{now} - 30 minutes).

This means that the authenticating certificate statement must be received every 30 minutes. Channels obeying freshness constraints are called recent-secure channels and are defined as follows:

Definition *A* . *B* **is** *recent-secure at time* t_{now} *iff A* . *B can be deduced from given statements with freshness constraints applied.*

It follows from the above definitions that if freshness constraints are associated with each trusted certifier whose statements may be used to establish a channel, then the delay for revocation can be bounded by the least restrictive freshness constraint. Consequently, the choice of freshness constraints can arbitrarily bound the delay for revocation. Certifiers recommend freshness constraints typically by imposing "notafter" times in certificates. However, freshness policies can be promulgated in varying types of certificates from whoever is specified as an authority as will be explained later. In addition, the authority for the policies can change depending on the nature of the transaction.

Kerberos is designed to enable application servers to impose freshness constraints on users who initially log-in using shared keys. To do this, the time of initial log-in "authtime" is carried in a ticket that is presented to the application server. Application servers can choose to reject the credentials if the "authtime" is not recent even if the ticket is still within its valid lifetime.

A recent proposal calls for public key extensions for initial authentication. With the exception of public-key authentication during initial log-in, all protocols remain the same. The implication of recent-secure authentication is that application servers need to know that their recent-secure authentication policies are satisfied. In the present invention, verifiers are employed to determine what policies need to be satisfied.

With the addition of public-key authentication to Kerberos, the current "authtime" field may not be sufficient for application servers to determine if initial authentication satisfies their authentication policies. The problem is complicated by the fact that recent-secure authentication policies may vary for each server, and possibly for each particular type of transaction. One approach to make Kerberos "recent-secure friendly" is to require users to satisfy prescribed recent-secure authentication policies prior to obtaining a ticket. During the course of a session, the application server may require the user to satisfy new policies or simply maintain freshness policies during a long session (e.g., refresh stale certificates). A new ticket can be issued once recent-secure authentication is satisfied.

Additionally, a revocation service should have the properties of being definite, available and recent, contained, adjustable and having bounded recovery.

Since large distributed systems have emerged, there has been a need for improving the joint authority technique (i.e., improving the technique of using a separate identification authority and revocation authority) for building a highly available and secure revocation service. The revocation service of the present invention uses identification authorities for issuing long-lived identification certificates and revocation authorities for issuing time stamped certificates. The time stamped certificates are stamped at date of issue (i.e., at the time when an entity became valid) do not require expiration dates. The identification authorities retain control of the specification of revocation certificates. Information about the freshness of revocation certificates is made explicit. That is, freshness constraints are determined from the time interval of verification.

The identification authority and the revocation authority have only unidirectional communication to the network. However, this approach can be complemented, particularly when revocation certificates must be extremely fresh and propagation delays in any conceivable directory are too slow. One approach is to use an on-line secret-sharing scheme having low latency and highly reliable communication between the secret-sharing sewers.

When authenticating entities are able to access the central service they can obtain fresh revocation certificates. A mechanism is needed for posting updates to the revocation service. This can include.

The identification authority specifies freshness constraints for revocation within the identification certificates. That is, in addition to conventional key and name binding the revocation authority specifies the restriction that revocation certificates obtained from the revocation authority meet designated freshness constraints. The significance of specifying freshness constraints in the identification certificates, delegating revocation authority, and issuing time stamped revocation certificates are as follows. Again, it should be noted that the freshness constraints can be specified in any type of certificate by any type of authority, not just by an identification authority and not in just an identification certificate.

The revocation authority need not be trusted to assign correct lifetimes to short-term revocation certificates. Otherwise, even after the compromised revocation authority is discovered, these certificates might still be used. In addition, clients need not interact with trusted on-line servers containing keys. It is preferable that they do not since this would compromise security. That is, high availability relies on data replication using untrusted servers instead of replicating trusted processes. The service is also secure because both identification and revocation authorities have unidirectional communication to the network.

Using time stamped revocation certificates instead of certificates with explicit expiration times scales better since the same time stamped certificates can be used by any authenticating entity and can be used for multiple freshness policies. Finally, availability is possible assuming the use of directories. Therefore, revocation according to the present invention requires less trust, is secure, is scalable and is highly available.

The statements of the revocation service will now be described with respect to policy constraints and assumptions as follows, and as set forth in the flow chart in Fig. 4B.

Entities who want to establish a channel, for example, *B*, require a secure channel with an identification authority *IA*. The entity must trust that IA has the authority to issue a certificate to *B*. This is established as follows.

(1) $K_{IA} \Rightarrow IA$

That is, the key for the identification authority K_{IA} speaks for the identification authority IA. Therefore,

(2) $IA \Rightarrow B$

That is, the identification authority *IA* speaks for *B*.

Then the off-line identification authority issues the following certificate.

(3) K_{IA} **says** $((K_B \Rightarrow B$ **notbefore** t_1 **notafter** $t_2) \wedge (\mathrm{RA}|K_B \Rightarrow B$ **notbefore** $(t_{now} - \delta_{IA \Rightarrow B})))$,

where K_B is the key for *B*, *RA* is the revocation authority and $\delta_{IA,\,B}$ is a recent-secure policy.

This statement has several important features. First, the expiration time of the certificate for *B* is long lived. Second, *IA* imposes its recent-secure policy, $\delta_{IA,\,B}$, on *B* using the constraint, $RA/K_{B},\ B$ **notbefore** $(t_{now} - \delta_{IA,\,B})$. This means that for K_B to be valid, claims made by RA/K_B must be interpreted according to U's freshness policy stated in this certificate.

The revocation authority *RA* states that the identification certificate is current.

(4) R/K_B **says** $(K_B, RA/K_B$ **notbefore** t_3 **notafter** $(t_3 + \delta_{IA,B}))$ **at** t_3. If the identification certificate is not current, i.e., the freshness constraint has expired, access is denied.

The particular value of $\delta_{IA,B}$ is not interpreted from the message corresponding to statement (4). Instead it is obtained from the original certificate statement (3).

It should be noted that more elaborate forms of joint authority might be useful. For example, using disjunction, the number of revocation authorities can be specified.

The above statements of revocation service will now be applied to a particular example. A basic certificate topology will be now described and analyzed with respect to Fig. 5A. In Fig. 5A, identification certificates are indicated by directed arrows shown in bold and revocation certificates are indicated by directed arrows shown by light lines from the certificate authority to the entity designated in the certificate. A delegation certificate is indicated by a dashed bold arrow. The reference ① refers to a distinguished principal. The distinguished principal delegates initial assertions ④ and ⑤ to objects of the assertion ② and ③. The distinguished principal indicated by ① also delegates through an assertion ⑥, authority for validity of initial assertions to an object ⑦. The object ⑦ is an object of delegation authority for asserting a time stamp and a validity of initial assertions. The object ⑦ then delegates an assertion about the time stamp and validity of an initial assertion ⑨ and 10 to the objects of the assertion ② and ③.

Fig. 5B is a certificate topology according to the present invention. A delegation certificate is indicated by a dashed bold arrow.

Policies for constraining the authorities of the principals can be expressed as initial statements which can be appropriately restrictive. For example,

$\boldsymbol{Org}\ .\ \boldsymbol{Org/^{*}},$

where * represents all identities. This can also be written as

$\boldsymbol{Org}\ .\ \boldsymbol{Org/Eve}.$

Hence, one may not necessarily believe that $\boldsymbol{Org}\ .\ \boldsymbol{Org/Joe}.$

Additionally, constraints can be expressed as formulas by augmenting or revising the specification to express valid formulas using the syntax described above. For example, negation ($\neg$) or conjunction ($\wedge$) can be used on formulas to express further restriction on the interpretation of messages.

Verification policies can also be enforced to modify the above-mentioned axioms. That is, the axioms can be further constrained by further constraining principals with particular principals. For example, the principal ***Eve*** can be replaced with ***Eve*** $\wedge$ ***Bob*** or replace ***Eve*** with ***Bob*** quoting ***Eve***, that is ***Bob/Eve***. Similarly, ***"notbefore"*** and ***"notafter"*** qualifiers can be made more restrictive. Also, times of statements using "at" can be made more restrictive by specifying particular times.

The authenticating entity's goal is to authenticate that public key $K_{Org/Eve}$ can be used to establish a secure channel with a principal *Org/Eve*. That is,

(1) $K_{Org/Eve}\ .\ Org/Eve.$

Then, for example, the "authenticating entity is required to obey the following freshness constraints:

(2) $\delta_{PCA\,.\,Org}$ = 31 days

(3) $\delta_{Org\,.\,Org/Eve}$ = 30 minutes.

The freshness constraints are dictated by the particular value of the transaction to be authenticated. Also, the authenticating entity obeys the constraints recommended by all trusted intermediaries. The time of authentication, for example, is:

(4) t_{now} = 14:00 6/15/96

Suppose that for a particular type of transaction the authenticating entity can only trust entities within a particular hierarchical security domain indicated by a top level policy certification authority (PCA). A security domain indicates that principals have compatible security policies. Also, the authenticating entity trusts the PCA to certify the parent organization of the principal that is to be authenticated. That is,

(5) *PCA* . *Org*

Also, it is assumed that the authentication entity is required to initiate the authentication chain starting from the top level *PCA* certificate due to, for example, desired security assurance and liability arrangements. Initially, it is assumed that the authenticating entity knows the top level key of the *PCA*. The top level key of the PCA is identified as follows:

(6) K_{PCA} . *PCA* **notafter** 00:00 8/01/96.

It will be assumed for brevity's sake that the certifying authorities (subordinate to the *PCA*) may only assign certificates using hierarchical names subordinate to their own names. That is,

(7) *PCA* . *PCA/REVOKE-CA*

(8) *Org* . *Org/Eve*

(9) *Org* . *Org/REVOKE-CA-PTR*

However, the policy within the security domain allows specially designated delegation certificates to indicate other trusted entities within the same security domain. That is, the delegation certificate specifies the name of an entity authorized to be recognized as an authority on issuing time stamped revocation certificates. An example is a medium-term delegation certificate indicated by the dashed arrow in Figs. 5A, 5B and 5C. When the authenticating entity is presented with, for example, a long tern certificate such as X.509, *Org* is interpreted as follows:

(10) K_{PCA} **says** (K_{Org} . *Org* **notafter** 00:00 1/1/97)

Also, the most recent monthly X.509 certificate revocation list is presented to the authenticating entity as:

(11) K_{PCA} **says** (K_{Org} . *Org* **notafter** at 00:00 1/1/97) **at** 6/1/96

The original CRL message need not repeat the field of the original certificate message.

Next, a special certificate for *Org/Eve* will be explained. The certificate specifies the key for *Org/Eve*, i.e., $K_{Org/Eve}$, and also designates *Org/REVOKE-CA-PTR* as an authority for revocation. For the particular type of transaction of interest, *Org* assumes liability only if a 30 minute recent revocation certificate statement has been obtained. That is:

(12) K_{Org} **says** (($K_{Org/Eve}$ ⇒ *Org/Eve* **notafter** 00:00 4/1/97) ∧ (*Org/REVOKE-CA-PTR*/$K_{Org/Eve}$ ⇒ *Org/Eve* **notbefore** (t_{now} - 30 minutes).

The authenticating entity is also presented with the most recent weekly certificate which specifies *Org/REVOKE-CA-PTR* as the entity *Org* trusts to serve as a revocation agent. In this example, *Org/REVOKE-CA-PTR* does not actually exist. Rather, it serves as delegation to a particular revocation agent. This technique enables flexibility in changing the revocation agent without having to reissue the long-term certificate. *Org* satisfies *PCA's* requirement that the designated entity be within the security domain. Of course, the agent may serve multiple security domains provided it satisfies the policy for each domain. The "notafter" suffix constrains exposure of the identification authority to future attacks on the revocation service. The effective bound, however, may extend to the least restrictive freshness constraint in any one of the identification certificates. That is,

(13) K_{Org} **says** (*PCA/REVOKE-CA* ⇒ *Org/REVOKE-CA-PTR* **notbefore** 00:00 7/1/96 **notafter** 00:00 7/20/96)

A longer certificate can be specified and a different revocation authority can be required to assert it's status. However, each level of indirection comes at an expense.

Also, the long term identification certificate for a central revocation service is given as

(14) K_{PCA} **says** ($K_{PCA/REVOKE-CA}$ ⇒ *PCA/REVOKE-CA* **notafter** 00:00 1/1/97).

The following revocation agent certifies that *Org/Eve's* certificate was current at 13:40 7/15/96:

(15) $K_{CPA/REVOKE\text{-}CA}/K_{Org/EVE}$ says ($K_{Org/Eve}$. $PCA/REVOKE\text{-}CA/K_{Org/EVE}$ **notbefore** 13:40 7/15/96 **notafter** (13:40 7/15/96 + $\delta_{Org.\ Org/Eve}$)) at 13:40 7/15/96.

The time stamp of the certificate message is interpreted as a "notafter" variable in the above statements (14) and (15). The value of $\delta_{Org.\ Org/Eve}$ is not made explicit but is interpreted from the original certificate statement (12) issued by *Org*.

The real certificate can contain one or more original certificates and a time stamp applied by PCA/REVOKE-CA. Alternatively, a condensed version might contain only a dated list of revoked certificates. Note that the agent can be made secure since real-time interaction with a trusted revocation authority is unnecessary for the purposes of this invention.

The primary processing function required by PCA/REVOKE-CA is the off-line function of dating and signing statements. Distribution of this information is made secure using unidirectional links to the directory. To prevent denial of service, these links must be highly reliable. Additionally, multiple revocation agents can be designated.

The certificate path will now be analyzed starting at the node *PCA* which is trusted by the authenticating entity. The lack of a time stamp in the X.509 certificate prevents a freshness determination from the certificate alone as shown in the statement (10). The CRL statement (11) is needed to determine freshness. Using the statements (5), (6), (10), (P6), (P7), (P8), and applying the transitivity rule for ⇒, the following is obtained:

(16) K_{Org} . *Org* **notafter** 00:00 1/1/97.

Statement (16) illustrates why the analysis is only good for the time of verification. Although the validity of statement (6) is good until 00:00 8/01/96, the conclusion is a fact that might wrongfully be interpreted as being valid until 00:00 1/1/97.

Using statements (6), (7), (14), (P6), (P7), (P8), and applying the transitivity rule for ⇒, the following is obtained:

(17) $K_{PCA/REVOKE\text{-}CA}$. *PCA/REVOKE-CA* **notafter** 00:00 1/1/97

Using statements (8), (12), (16), (P7) and (P8) followed by normalizing the suffix constraint using (P6), the following is obtained:

(18) ($K_{Org/Eve}$. *Org/Eve* **notbefore** (t_{now} - 30 minutes) **notafter** 00:00 4/11/97) ∧ (*Org/REVOKE-CA-PTR*|$K_{Org/Eve}$. *Org/Eve* **notbefore** (t_{now} - 30 minutes) **notafter** 00:00 4/1/97).

Using statements (9), (13), (16), (P7) and (P8) the following is obtained:

(19) *PCA/REVOKE-CA* . *Org/REVOKE-CA-PTR* **notbefore** 00:00 7/13/96 **notafter** 00:00 7/20/96.

Using statement (P6) to normalize the suffix times on statements (18) and (19) and then applying the transitivity rule for ⇒, the following is obtained:

(20) ($K_{Org/Eve}$. *Org/Eve* **notbefore** 13:30 7/15/96 **notafter** 00:00 7/20/96) ∧ (*PCA/REVOKE-CA*|$K_{Org/Eve}$ ⇒ *Org/Eve* **notbefore** 13:30 7/15/96 **notafter** 00:00 7/20/96).

Using statements (15), (17), (P7) and (P8) the following is obtained:

(21) $K_{Org/Eve}$. *PCA/REVOKE-CA*|$K_{Org/Eve}$ **notbefore** 13:40 7/15/96 **notafter** (13:40 7/15/96 + $\delta_{Org/Eve}$).

Statements (21) and (20) are normalized using (P6) and the transitivity rule for ⇒ is applied. Like terms are then combined to obtain:

(22) $K_{Org/Eve}$. *Org/Eve* **notbefore** 13:40 7/15/96 **notafter** 14:10 7/15/96.

Since 13:40 7/15/96 $\leq$ (t_{now} 14:00 6/15/96) $\leq$ 14:10 7/15/96, it can be concluded that

(23) $K_{Org/Eve}$. *Org/Eve*

Fig. 5C is a diagram of a certificate topology according to a second embodiment of the present invention. Fig. 5C is similar to the embodiment in Fig. 5B except that the PCA is not present. in this case, the initial delegating assertion is issued by the first entity to a second entity. The second entity delegates authority to a third entity.

Fig. 6 is a diagram of a replicated directory organized for the propagation of certificates according to the recent-secure delay bounds of authenticating entities. That is, it is a replicated directory with varying levels of persistent storage. This type of directory can be used with any type of certificate topology according to the present invention. It need not be assumed that the directory is trusted. The High level includes items that are replicated frequently including time stamped assertions (i.e., revocation certificates or validity certificates). The Medium level includes items that are replicated often including the time stamped assertions and medium-term delegation assertions. The Low level includes items that are replicated infrequently including the time stamped assertions, the medium-term delegation assertions and long-term identification assertions.

The storage area of the present invention need not be limited to the above example. Rather, the storage area can include a storage system or communications network. The communications network can be a multicast address group.

Figs. 5A, 5B, 5C, and 6 show that recent-secure authentication uses freshness constraints imposed by the authenticating entity and constraints recommended by intermediary certifiers. The recent-secure authentication of the present invention uses a technique for delegating revocation authority without requiring that the revocation agent be trusted to specify revocation policies. It is assumed that the statements from each channel principal can be ordered and missing statements can be deleted. Also, as set forth above, the order of statements from different sources can be established using clock synchronization. The external synchronization accuracies can also be determined. The recent-secure authentication of the present invention also provides flexibility for changing revocation agents using indirection in long-term certificates. Further, the addition of time stamps to, for example, X.509, may improve the performance for satisfying freshness constraints in specialized architectures in a similar manner as identification assertions.

System specification is a preliminary analysis step. Analysis includes the steps of expressing the channel verification goal in terms of language formulas, deriving new statements by applying statement and principal axioms to system specification statements subject to policy constraints and concluding analysis when a new derived statement achieves a verification goal or terminating analysis.

The present invention provides specifying security protocols and policy constraints for establishing secure channels in distributed systems. It also includes a method and system for refined specification of policies for revocation and validity having a bounded delay. The revocation and validity policies are embodied in freshness constraints imposed on recent-secure (channel) authentication. The present system and method for reasoning about recent-secure authentication and for promulgating revocation policies and validity policies in authenticated statements enables the design of a secure and highly available system for revocation where less trust is required of the revocation and validity service.

Additionally, the present invention enables authenticating entities' on a per transaction basis, to adjust authentication costs at the expense of protection. This is because the present invention makes precise what degree of protection is desired and whether that protection is obtained. Formal recent-secure policies provide a basis for both optimizing the distribution of credentials, e.g., public key identification, revocation and validity assertions, and for designing authentication architectures that are resilient to network partitioning or disconnected operations.

The present invention can also be extended to include a "received notafter" suffix to signal that the integrity of statements received after a specified time has an unacceptably high threshold of being compromised. Message contents cached prior to integrity timeouts, however, might still be usable. Such techniques enable cached assertions to be used past the stated time periods and enable computational efficiencies due to choices in key sizes.

The foregoing is considered as illustrative only of the principles of the present invention. Further, since numerous modifications will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and applications shown and described, and accordingly, all suitable modifications and equivalents may be resorted to, falling within the scope of the invention and the appended claims and their equivalents.